VisTrails Provenance Traces for Benchmarking

Fernando Chirigati
Polytechnic Institute of NYU
fchirigati@nyu.edu

David Koop
Polytechnic Institute of NYU
dkoop@poly.edu

Juliana Freire
Polytechnic Institute of NYU
juliana.freire@nyu.edu

Cláudio Silva
Polytechnic Institute of NYU
csilva@nyu.edu

1. INTRODUCTION

The benchmark provenance traces that we have collected come from the VisTrails system for exploratory data analysis and visualization and from the VisTrails, Inc. provenance plugin for Autodesk Maya [1]. They contain different kinds of provenance information, including prospective and retrospective provenance as well as provenance of the evolution of workflows and models [4]. Our traces are stored using a *change-based* representation to compactly save all directions a user explored when developing a result.

2. CHANGE-BASED PROVENANCE

The change-based model was introduced with the VisTrails system to maintain a set of application states as a tree of actions [5]. Instead of storing these states or *versions* individually, the change-based model records the actions that transform one state to another. Not only does this require less space than storing multiple versions, but the actions can also be higher-level and more informative than the computed differences version control systems employ. For example, a transformation on a mesh might be captured as a single command with some parameters rather than the perturbations of each point in the mesh. Furthermore, the implementation used in both VisTrails and the VisTrails plugins automatically captures changes—a user does not need to explicitly define each version, although they may add a tag or other annotations. A tree-based view of this provenance allows users to not only undo and redo steps but also jump to any version previously created.

3. THE VISTRAILS SYSTEM

VisTrails (`http://www.vistrails.org`) is an open-source provenance management and scientific workflow system designed to support the scientific discovery process [3, 6]. It provides support for data analysis and visualization, together with a user-centered design. The system combines and substantially extends useful features of visualization and scientific workflow systems, enabling users to create complex workflows that encompass important steps of scientific discovery, from data gathering and manipulation, to complex analyses and visualizations, all integrated in a single system.

A key feature of VisTrails is its *comprehensive provenance infrastructure* that maintains detailed history information about the steps followed and data derived in the course of an exploratory task [5]—VisTrails maintains provenance of data products, of the workflows that derive these products, and of their executions. The workflow evolution and workflow specifications are stored using change-based provenance. The provenance helps users to reason about the results, follow chains of reasoning backward and forward, and explore workflow versions in an intuitive way, using a history tree. Because of the change-based provenance, users do not lose any results, even when undoing changes.

4. THE VISTRAILS PROVENANCE SDK

The change-based provenance used in the VisTrails workflow system has also been generalized to support applications that follow the model-view-controller paradigm [2]. VisTrails, Inc., has developed the Provenance Software Development Kit (ProvSDK) to capture the evolution of all results for an application [8]. Each application defines methods for serializing and deserializing actions, and the SDK takes care of metadata, version dependencies, and interfaces for search, playback, and inspection.

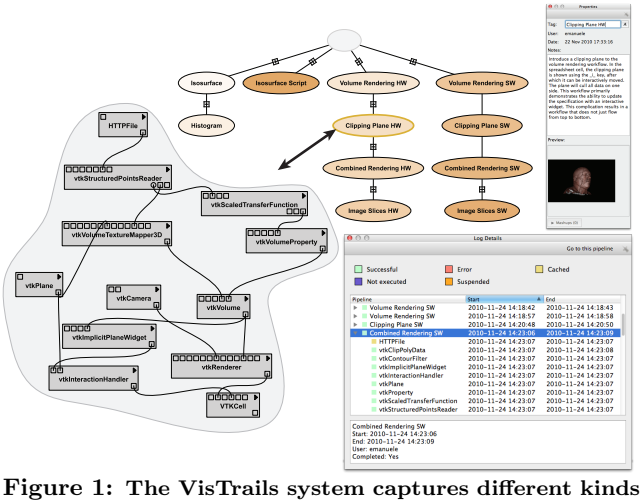

Figure 1: The VisTrails system captures different kinds of provenance. The version tree captures the workflow evolution and allows users to navigate over past workflow versions. In addition to automatically captured metadata, users may also add annotations to each version. The system also records run-time information in the execution log.

EDBT/ICDT '13, Mar 18-22 2013, Genoa, Italy

5. PROVENANCE TRACES

The provided traces are from the scientific and information visualization domain, and they encompass the three types of provenance captured by VisTrails: prospective, retrospective and workflow evolution. We include workflows that, for instance, read structured and unstructured grid data, extract an isosurface from a model and render surfaces and volumes. Additionally, we include some provenance traces generated in the VisTrails plugin for Autodesk Maya [1], which uses an early version of the VisTrails Provenance SDK [8] to transparently capture the provenance of the user's actions when building three-dimensional models. Table 1 provides a summary of the included traces.

Data Format	XML
Data Model	VisTrails native schema
Size	5.2 MB
Tools	VisTrails system and ProvSDK
Application Domain	Visualization
Submission Group	Refer to authors and affiliation
Contact	Refer to authors and affiliation
License	cc-by-nc-sa [7]

Table 1: Summary of the VisTrails provenance traces

Provenance Queries. Below are some possible provenance queries that can be evaluated using our provided traces.

- What was the set of parameters used in module m?
- How many times was module m executed?
- How long did the execution of module m take?
- To which module m is module execution m' related?
- In which workflow version v was module m added?
- In which workflow version v was parameter p was set?
- To which workflow version v is module execution m' related?
- From which version v was version v' derived?
- When did user u last modify version v?

Coverage of PROV. Some of the VisTrails native schema terms correspond to the PROV data model. In fact, there is a translation from the VisTrails schema to PROV, and VisTrails provides a serialization to XML (PROV-XML). Table 2 presents the coverage of PROV in VisTrails.

6. CONCLUSION

The ability to navigate through different versions and compare them, never losing previous results, is one of the key features of VisTrails and the ProvSDK, and the provenance traces contain information with respect to the workflow evolution. Applications interested in keeping track of all user's actions can directly benefit from the submitted provenance traces by looking at how VisTrails systematically stores the workflow versions. In addition, the change-based provenance contained in the traces present the opportunity to explore a form of provenance that is different from most other submissions.

PROV-O Term	Covered?
prov:Activity	Y
prov:Agent	Y
prov:Entity	Y
prov:actedOnBehalfOf	N
prov:endedAtTime	Y
prov:startedAtTime	Y
prov:used	Y
prov:wasAssociatedWith	Y
prov:wasAttributedTo	N
prov:wasDerivedFrom	N
prov:wasGeneratedBy	Y
prov:wasInformedBy	N

Table 2: Coverage of PROV in VisTrails

7. REFERENCES

[1] Maya. http://usa.autodesk.com/maya/.

[2] S. P. Callahan, J. Freire, C. E. Scheidegger, C. T. Silva, and H. T. Vo. Towards provenance-enabling paraview. In *IPAW*, pages 120–127, 2008.

[3] J. Freire, D. Koop, E. Santos, C. Scheidegger, C. Silva, and H. T. Vo. *The Architecture of Open Source Applications*, chapter VisTrails. Lulu.com, 2011.

[4] J. Freire, D. Koop, E. Santos, and C. T. Silva. Provenance for computational tasks: A survey. *Computing in Science and Eng.*, 10(3):11–21, May 2008.

[5] J. Freire, C. Silva, S. Callahan, E. Santos, C. Scheidegger, and H. Vo. Managing rapidly-evolving scientific workflows. In *International Provenance and Annotation Workshop (IPAW)*, LNCS 4145, pages 10–18. Springer Verlag, 2006.

[6] J. Freire and C. T. Silva. Making computations and publications reproducible with vistrails. *Computing in Science and Engineering*, 14(4):18–25, 2012.

[7] Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. `http://creativecommons.org/licenses/by-nc-sa/3.0/`.

[8] VisTrails Provenance SDK. `http://www.vistrails.com/sdk.html`.